02

Conociendo la interface List

Transcripción

[00:00] Hola. ¿Cómo están? Vamos a iniciar nuestra clase 6. Nuestra clase 6 va a ser conocer un poco de la interface list y sus diferentes implementaciones. Para eso primero aquí hice un renaming de nuestro package com.Alura. Voy a crear un nuevo package y voy a colocarle model. Entonces en model vamos a colocar nuestra clase curso. Además, vamos a duplicar esta clase curso y vamos a colocarle clase.

[00:54] Toda clase, vamos a tener un nombre de la clase que vamos a recibir y el tiempo también de cada clase. O solamente vamos a utilizar el nombre, vamos a hacerlo más simple. Una vez hecho esto, aquí en el curso vamos a decir que tenemos una lista, ¿de qué? De clases. Entonces para eso vamos a utilizar aquí private List Clase y vamos a colocar aquí claseList.

[01:43] ¿Aquí qué estamos haciendo? Estamos utilizando la interface list. Eso quiere decir, ya no estamos utilizando la ArrayList que sería una clase. Aquí es una interface. Esto quiere decir que aquí, en interface list, de acuerdo con la documentación de Java podemos implementar un ArrayList, un LinkedList, un vector o varios otros tipos de implementaciones.

[02:10] Eso nos va a ayudar a hacer una clase más genérica para después nosotros poder actualizarla. Por ejemplo, vamos a utilizar esta lista como un ArrayList. Después si queremos podemos hacerle un nuevo nombre, le podemos cambiar para LinkedList o RoleList dependiendo de la necesidad. Esas son unas ventajas de trabajar con interfaces. Eso es lo que se le puede

llamar polimorfismo, o sea podemos cambiar la clase dependiendo de las implementaciones.

[02:45] Por ejemplo, tenemos aquí list y podemos al inicio trabajar con ArrayList. Después podemos cambiarlo por LinkedList. Vamos a hacer lo siguiente. Aquí vamos a colocarle new ArrayList. Una vez hecho esto, podemos crear dos constructores, uno para nombre tiempo y otro para nombre tiempo y un list de clase: this.claseList = claseList.

[03:34] Vamos a implementar nuestros métodos getters and setters. Una vez hecho esto, vamos a implementar un método para adicionar a nuestra lista. Voy a utilizar un public void addClase. Aquí vamos a recibir un objeto del tipo clase y vamos a adicionar: claseList.add(clase);.

[04:20] Esto nos va a facilitar para después dentro de este método ya podemos hacer diferentes modificaciones y ya no tendríamos que ir al método principal para después alterar o hacer alguna tipo de modificación. Aquí por ejemplo, me faltó el punto y coma. Y ahora vamos aquí a duplicar nuestra clase 5 y vamos a colocar clase 6.

[04:52] Nuestra clase 6, tenemos aquí varios objetos. Vamos a remover todo lo que está aquí y listo. Aquí tenemos una lista, por ejemplo, vamos a crear una lista de objeto curso. Una vez hecho esto vamos a remover aquí y después en curso1 vamos a colocar aquí new ArrayList por ejemplo. Estamos instanciando.

[05:36] Una vez hecho esto, tenemos aquí la lista de curso, y en curso1 vamos a colocar así: curso1.addClase. Vamos a colocar new Clase. En este new Clase vamos a ver, aquí tenemos, perfecto, podemos colocar un nombre de la clase. Vamos a colocarle.

[06:06] En el curso de Java, vamos a colocarle que la clase va a ser de ArrayList y después vamos adicionar. Una vez hecho esto, aquí podemos imprimir lista de cursos, punto, vamos al índice 0, que sería el primer objeto de la lista, .getClaseList y vamos a ejecutar.

[06:52] ¿Apareció qué cosa? ArrayList que fue lo que aquí adicionamos. Entonces, aquí lo que estamos viendo en la clase 6, es utilizar la clase curso para adicionar una lista dentro de esa clase. Una vez hecho esto, hemos adicionado una clase llamada ArrayList y después hemos impreso esa misma clase.

[07:25] Por ejemplo aquí podemos adicionar otra clase más, no va a ser ArrayList. Después vamos a utilizar List. Por ejemplo vamos a tener otra clase para LinkedList, y ahora vamos a ejecutar y tenemos ArrayList, List y LinkedList. La parte importante es que nosotros estamos utilizando la interface list para declarar una lista del tipo ArrayList.

[08:05] Una vez hecho esto, por ejemplo, aquí podemos cambiar, aquí no era necesario poder utilizar el ArrayList. ¿Por qué? Porque una vez que estamos creando ya estamos diciendo que esto de aquí va ser ArrayList, nuestra clase list. Entonces aquí vamos a imprimir. Y va a dar lo mismo. ¿Cuál es la ventaja?

[08:32] Que aquí podemos ponerle otro tipo de objeto. Vamos a colocar un LinkedList, y automáticamente toda nuestra lista se convirtió en LinkedList. Eso es lo que se llama polimorfismo. Entonces espero que haya sido de su agrado, nos vemos en la siguiente clase y muchas gracias.